Hindawi
BioMed Research International
Volume 2017, Article ID 4590609, 10 pages
https://doi.org/10.1155/2017/4590609

Research Article

HMMBinder: DNA-Binding Protein Prediction Using HMM Profile Based Features

Rianon Zaman,[1] Shahana Yasmin Chowdhury,[1] Mahmood A. Rashid,[2,3] Alok Sharma,[3,4,5] Abdollah Dehzangi,[6] and Swakkhar Shatabda[1]

[1]*Department of Computer Science and Engineering, United International University, Dhaka, Bangladesh*
[2]*School of Computing, Information and Mathematical Sciences, The University of the South Pacific, Suva, Fiji*
[3]*Institute for Integrated and Intelligent Systems, Griffith University, Brisbane, QLD, Australia*
[4]*School of Engineering and Physics, The University of the South Pacific, Suva, Fiji*
[5]*RIKEN Center for Integrative Medical Sciences, Yokohama, Japan*
[6]*Department of Computer Science, Morgan State University, Baltimore, MD, USA*

Correspondence should be addressed to Swakkhar Shatabda; swakkhar@cse.uiu.ac.bd

Received 29 August 2017; Accepted 22 October 2017; Published 14 November 2017

Academic Editor: Paul Harrison

DNA-binding proteins often play important role in various processes within the cell. Over the last decade, a wide range of classification algorithms and feature extraction techniques have been used to solve this problem. In this paper, we propose a novel DNA-binding protein prediction method called HMMBinder. HMMBinder uses monogram and bigram features extracted from the HMM profiles of the protein sequences. To the best of our knowledge, this is the first application of HMM profile based features for the DNA-binding protein prediction problem. We applied Support Vector Machines (SVM) as a classification technique in HMMBinder. Our method was tested on standard benchmark datasets. We experimentally show that our method outperforms the state-of-the-art methods found in the literature.

1. Introduction

DNA-binding proteins play a vital role in various cellular processes. They are essential in transcriptional regulation, recombination, genome rearrangements, replication, repair, and DNA modification [1]. Proteins which make bond with DNA in both eukaryotes and prokaryotes while performing like activators or repressors are DNA-binding proteins. It has been observed that the percentages of prokaryotes and eukaryotes protein that can bind to DNA are only 2-3% and 4-5%, respectively [2, 3]. There have been a wide variety of experimental methods such as *in vitro* methods [4, 5] like filter binding assays, chromatin immunoprecipitation on microarrays (ChIP-chip) genetic analysis, and X-ray crystallography which are used to predict DNA-binding proteins. However, these methods are proven to be expensive and time consuming. Therefore, there is a growing demand to find a fast and cost effective computational method to solve this problem.

Most of the computational methods used in the literature to predict DNA-binding proteins formulated the problem as a supervised learning problem. Practically, the number of known DNA-binding proteins is very small compared to the large non-DNA-binding proteins and unknown proteins. DNA-binding protein prediction is often modeled as a binary class classification problem where given a protein sequence as input the task is to predict whether the protein is DNA-binding or not. Note that the challenge here is to select a proper dataset for training and testing incorporating the imbalanced situation. Many supervised learning algorithms have been used in the literature to solve the problem. Among them, Artificial Neural Networks (ANN) [6], Support Vector Machines (SVM) [7, 8], ensemble methods [9], Nave Bayes classifier [10], Random Forest [11], Convolutional Neural Networks [12], Logistic Regression [13], AdaBoost Classifier [5], and so on are well-regarded. Support Vector Machines (SVM) are one of the best performing classifiers used for DNA-binding protein identification [7, 8, 14, 15].

A great number of web based tools and methods are developed for DNA-binding protein prediction and are available for use. In this paper, we would like to mention several of them: DNABinder [7], DNA-Prot [16], iDNA-Prot [11], iDNA-Prot|dis [14], DBPPred [17], iDNAPro-PseAAC [8], PseDNA-Pro [18], Kmer1 + ACC [19], Local-DPP [20], SVM-PSSM-DT [21], PNImodeler [22], CNNsite [12], and BindUP [23]. Most of these methods have used sequence, profile, or structure based features. In structural feature based methods in the literature, features used were structural motifs, electrostatic potential, the dipole moment, and α-carbon only models [13, 24, 25]. On the other hand, sequence based methods often depended on the PSSM profile based information or pseudo-amino-acid compositions [8, 14, 15, 17, 20, 26, 27]. In [28], HMM based profiles were used for generating features for protein fold recognition.

In this paper, we propose HMMBinder, a novel DNA-binding protein prediction tool using HMM profile based features of a protein sequence. Our method uses monogram and bigram features derived from the HMM profile which shows effectiveness compared to the PSSM or sequence based features. We also use SVM as the classifier and standard benchmark datasets to test our method. Using the standard evaluation metrics, our method significantly improves over the state-of-the-art methods and the features used in the literature. We also developed a web server that is publicly available at http://brl.uiu.ac.bd/HMMBinder.

The rest of the paper is organized following the general 5-step guideline suggested in [29] for protein attribute prediction. First, benchmark datasets selected for this problem are described followed by a description of the protein representation by extraction of features. Then we describe the classification algorithm that we selected for our approach followed by the performance evaluation techniques deployed in this paper. Lastly, we describe the web server that we developed for this problem. The results section presents the details of the experimental results followed by an analytical discussion. The paper concludes with a summary and indication of future work.

2. Methods and Materials

In this section, we provide the details of the materials and methods of this paper. Figure 1 provides a system diagram of our proposed method. For the training phase, all the protein sequences are fed to HHBlits [30], a sequence-to-sequence alignment software using the latest UniProt database. HHBlits produces HMM file as output which is then used by our feature extraction method to generate monogram and bigram features. Monogram and bigram features are concatenated together and then used as training feature set to train the classifier. We use SVM with linear kernel as the classification algorithm and a trained model is stored for the testing phase. Testing phase is also similar to the training phase; however, the labels for the test dataset are not given to the classifier. This stored model is also used for the web server implementation of HMMBinder.

2.1. Datasets. Selection of benchmark datasets is essential in classification and prediction design. In this paper we use a popular benchmark dataset called *benchmark1075* to train our model. Later we test the performance using cross validation and on a separate independent test set known as *independent186* dataset. This section provides a brief overview of these two datasets. Both of these datasets are widely used in the literature of DNA-binding protein prediction literature [8, 14, 18, 20, 31].

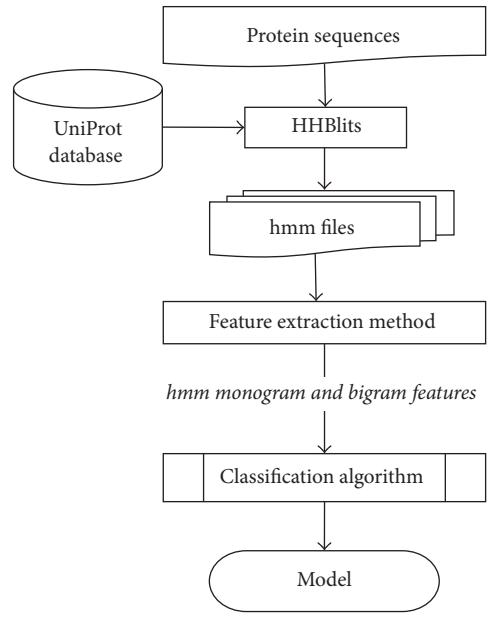

Figure 1: System diagram of HMMBinder.

2.1.1. Dataset Benchmark1075. This dataset was first introduced in [14]. This dataset consists of 1075 protein sequences. Among them, 525 are DNA-binding and 550 are non-DNA-binding protein sequences. All the protein sequences were taken from PDB [32]. This dataset is one of the largest DNA-binding protein prediction datasets and thus suitable for training purpose.

2.1.2. Dataset Independent186. Lou et al. [17] constructed this independent dataset consisting of 93 DNA-binding and 93 non-DNA-binding protein sequences. They used BLASTCLUST [33] on the benchmark dataset to remove the sequences that have more that 25% of similarity.

2.2. Feature Extraction. The training dataset $\mathbb{S}$ used for a binary classification problem consists of two types of instances: positive and negative. Formally,

$$\mathbb{S} = \mathbb{S}^+ \cup \mathbb{S}^-. \quad (1)$$

Next, the task is to represent each protein instance as feature vectors suitable for training. The idea is to represent each of the protein instances as a vector of features.

$$P = [f_1, f_2, \ldots, f_n]. \quad (2)$$

Here, a protein, $P \in \mathbb{S}$, is shown as a feature vector with dimension n. Most of the methods in the literature of DNA-binding protein prediction use either sequence and

PSSM profile based features or structure based features. To the best of our knowledge, there has been no application of features using HMM profiles. In this paper, we have used HHBlits [30] to generate HMM profiles. HMM profiles are comparatively more effective [30, 34] for remote homology detection. HMM profiles were generated using four iterations of HHBlits with a cutoff value set to 0.001 using the latest UniProt database [35]. HMM profiles are $L \times 20$ matrix produced by HHBlits. These 20 values are the substitution probability of each type of amino-acid residue along the protein sequence at each position. These values are first converted to linear probabilities using the following formula:

$$p = 2^{-N/1000}. \tag{3}$$

We generated two types of features, monogram and bigram, using the generated HMM profile matrix noted here as H. We provide a brief description of monogram and bigram features extracted from the HMM profile matrix.

2.2.1. Monogram Features. Monogram features [36] are calculated taking the normalized sum of the column wise substitution probability values. Size of these feature group is 20 because of 20 different amino acids. The feature can be defined formally as follows:

$$\text{HMM-Monogram}\left(j\right) = \frac{1}{L}\sum_{i=1}^{L} H_{i,j}. \tag{4}$$

Note that values of j are dependent on the columns; that is, $1 \le j \le 20$. Here, H_{ij} are the values in the ith row and jth column of the matrix. We denote monogram features as M which is a vector of the form $M = [m(1), m(2), \ldots, m(20)]$.

2.2.2. Bigram Features. Bigram features have been successfully used in the literature for protein attribute prediction [37]. Bigram features are normalized bigrams taken for all pairs of columns. Hence the total number of features generated from this group is 400. Bigram features are generated using the following formula:

$$\text{HMM-Bigram}\left(j,k\right) = \frac{1}{L-1}\sum_{i=1}^{L} H_{i,j}H_{i+1,k}. \tag{5}$$

Here j and k denote the column pairs for which the bigram is calculated and are in the ranges $1 \le j \le 20$ and $1 \le k \le 20$. We denote this feature vector as B, where B has the form of $B = [b(1,1), b(1,2), \ldots, b(1,20), b(2,1), \ldots, b(20,20)]$.

We also generate Positive Specific Scoring Matrix (PSSM) profiles for each of the protein sequences using PSI-BLAST [38]. PSSMs were generated using three iterations of PSI-BLAST using the nr database with a cutoff value of 0.001. PSSM profiles also have a similar form to HMM profiles which is a matrix of the same dimension and each of the matrix values denotes substitution probabilities. We generate monogram and bigram features from PSSM files as well. These PSSM based monogram and bigram features are well used in the literature [36, 37, 39–42]. Note that all the monogram features are vectors of size 20 and bigram features are vectors of size 400. We have also used a combination of the monogram and bigram features which is a vector of size 420.

2.3. Support Vector Machine. We have used Support Vector Machines (SVM) as our classification technique. SVM is successfully used in protein attribute prediction in general [28, 39, 43] and particularly in DNA-binding protein prediction [7, 8]. SVM is maximum margin classifier that attempts to learn a hyperplane from the training samples that separates the positive and negative data points in a binary classification problem. The hyperplane that is selected is the one for which the separation width or the margin is maximum and the nature of the hyperplane depends on the kernel functions used. SVM generally tries to optimize a multiplier function that goes as follows:

$$L = \arg\max_{\alpha} \sum_{j} \alpha_j - \frac{1}{2}\sum_{j,k} \alpha_j \alpha_k y_j y_k \phi\left(\vec{x}_j \cdot \vec{x}_k\right). \tag{6}$$

The prediction of a SVM classifier is defined as follows:

$$h\left(\vec{x}\right) = \text{sign}\left(\sum_{j} \alpha_j y_j \left(\vec{x} \cdot \vec{x}_j\right) - b\right). \tag{7}$$

Here the transformation of the data points by the function ϕ could be linear, polynomial, or any other kernel functions. In this paper, we explored linear and radial basis function (RBF) kernels. Linear kernel is of the following form:

$$K\left(\vec{x}_i, \vec{x}_j\right) = \left(\vec{x}_i \cdot \vec{x}_j + c\right)^d. \tag{8}$$

Here $d = 1$ for the linear kernels. RBF kernels follow the following definition:

$$K\left(\vec{x}_i, \vec{x}_j\right) = e^{(-)(\|\vec{x}_i - \vec{x}_j\|^2/2\sigma^2)}. \tag{9}$$

Often slack variables are used along with the maximum margin SVM classifier to allow generalization error depending on a parameter C.

2.4. Performance Evaluation. A good number of effective evaluation metrics have been suggested for use in single valued and multivalued classification and prediction [29, 44]. In the literature of DNA-binding protein prediction, we have found that the most widely used metrics are accuracy, sensitivity, specificity, MCC, auROC, and auPR values. In this section, we first provide a description of these evaluation metrics used in this paper.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{10}$$

This first measure, accuracy, is the ratio or percentage of correctly classified negative or positive instances from a given number of protein instances. Here TP is the total number of true positives or correctly classified positive samples and TN is the correctly classified negative samples. FP and FN are incorrectly classified positive and negative instances, respectively. Sensitivity is the true positive rate or the ratio of true positives to the total number of positive examples. Sensitivity is defined in the following equation:

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{11}$$

Table 1: Comparison of performances of different features and SVM kernels on the benchmark dataset using 10-fold cross validation.

Features	Accuracy	Sensitivity	Specificity	auPR	MCC	auROC
SVM with linear kernel						
HMM-Monogram	76.77%	**0.8420**	0.6976	0.6931	0.5367	0.8358
PSSM-Monogram	74.74%	0.6636	0.8362	0.8368	0.5040	0.8105
HMM-Bigram	70.59%	0.7071	0.7049	0.7060	0.4095	0.7511
PSSM-Bigram	62.20%	0.6454	0.5973	0.6025	0.2502	0.6703
HMM (Mono + Bi)	**82.87%**	0.8150	**0.8415**	**0.8428**	**0.6538**	**0.8639**
PSSM (Mono + Bi)	72.40%	0.7364	0.7120	0.7136	0.4486	0.8028
SVM with RBF kernel						
HMM-Monogram	**78.83%**	**0.8227**	0.7559	0.7535	**0.5761**	**0.8667**
PSSM-Monogram	73.71%	0.6890	0.7880	0.7903	0.4771	0.8121
HMM-Bigram	76.68%	0.7052	**0.8251**	0.8253	0.5283	0.8318
PSSM-Bigram	74.92%	0.7490	0.7495	0.7516	0.4966	0.8166
HMM (Mono + Bi)	77.43%	0.7129	0.8324	**0.8329**	0.5440	0.8496
PSSM (Mono + Bi)	72.40%	0.7363	0.7120	0.7136	0.4486	0.8028
Random Forest						
HMM-Monogram	**74.44%**	**0.7938**	**0.6976**	**0.6936**	**0.4871**	0.8243
PSSM-Monogram	66.14%	0.7290	0.5895	0.5862	0.3173	0.7332
HMM-Bigram	72.19%	0.7553	0.6903	0.6880	0.4400	**0.8273**
PSSM-Bigram	71.00%	0.7854	0.6300	0.6305	0.4174	0.7833
HMM (Mono + Bi)	74.43%	**0.7938**	**0.6976**	0.6931	**0.4871**	0.8218
PSSM (Mono + Bi)	72.68%	0.7909	0.6589	0.6645	0.4557	0.7698
AdaBoost						
HMM-Monogram	73.31%	0.7013	0.7632	0.7603	0.4579	0.8026
PSSM-Monogram	67.07%	0.7654	0.5703	0.5737	0.3448	0.7157
HMM-Bigram	73.97%	0.7360	0.7432	0.7396	0.4762	0.8063
PSSM-Bigram	70.53%	0.7436	0.6647	0.6708	0.4116	0.7710
HMM (Mono + Bi)	**78.00%**	**0.7803**	**0.7795**	**0.7732**	**0.5532**	**0.8577**
PSSM (Mono + Bi)	70.07%	0.7327	0.6666	0.6687	0.4005	0.7887

Specificity on the other hand is the true negative rate and can be defined as the following equation:

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}. \tag{12}$$

All these three measures have a maximum value of 1 which is the best classifier and a minimum value of 0 meaning the worst classifier. Mathew's Correlation Coefficient (MCC) denotes how good a binary classification is working. The value of MCC is in the range $[-1, +1]$. A perfect classifier should have a maximum MCC value of +1. MCC is defined as the following equation:

$$\text{MCC} = \frac{(\text{TP} \times \text{TN}) - (\text{FP} \times \text{FN})}{\sqrt{(\text{TP} + \text{FP})\,(\text{TP} + \text{FN})\,(\text{TN} + \text{FP})\,(\text{TN} + \text{FN})}}. \tag{13}$$

Note that all these metrics for probabilistic outputs depend on the threshold set for the classifiers. Two other metrics not dependent on thresholds are area under receiver operating characteristic curve (auROC) and area under precision-recall curve (auPR). The value of auROC and auPR has maximum value of 1 for the perfect classifier. ROC curve plots true positive rate against false positive rate at different threshold values and precision-recall curve plots precision against recall.

To reduce the training bias, several sampling methods are proposed in the literature [45] and widely used for protein attribute prediction [29]. In this paper, we have used 10-fold cross validation and jack-knife tests which are widely used in the literature of DNA-binding protein prediction [8, 11, 14, 17].

3. Results and Discussion

In this section, we present the results of the experiments that were carried out in this study. All the methods were implemented in Python3.4 programming language. The Scikit-learn library [46] of python was used for implementing the machine learning algorithms. All experiments were conducted on computing services provided by CITS, United International University.

3.1. Effect of HMM Based Features. We have run a number of experiments to test the effectiveness of the HMM profile based features on the benchmark dataset. We have six groups of features extracted for this experiment: PSSM-Monogram, PSSM-Bigram, PSSM-Mono + Bigram, HMM-Monogram, HMM-Bigram, and HMM-Mono + Bigram. Each of these feature sets is tested with SVM classifiers using linear and RBF kernels. We further tested the performances of these features using two ensemble classifiers: Random Forest and AdaBoost Classifiers. For these experiments we have performed 10-fold cross validation. The results in terms of accuracy, sensitivity, specificity, auPR, auROC, and MCC are reported. Only the average of these values is reported in Table 1. Note that the

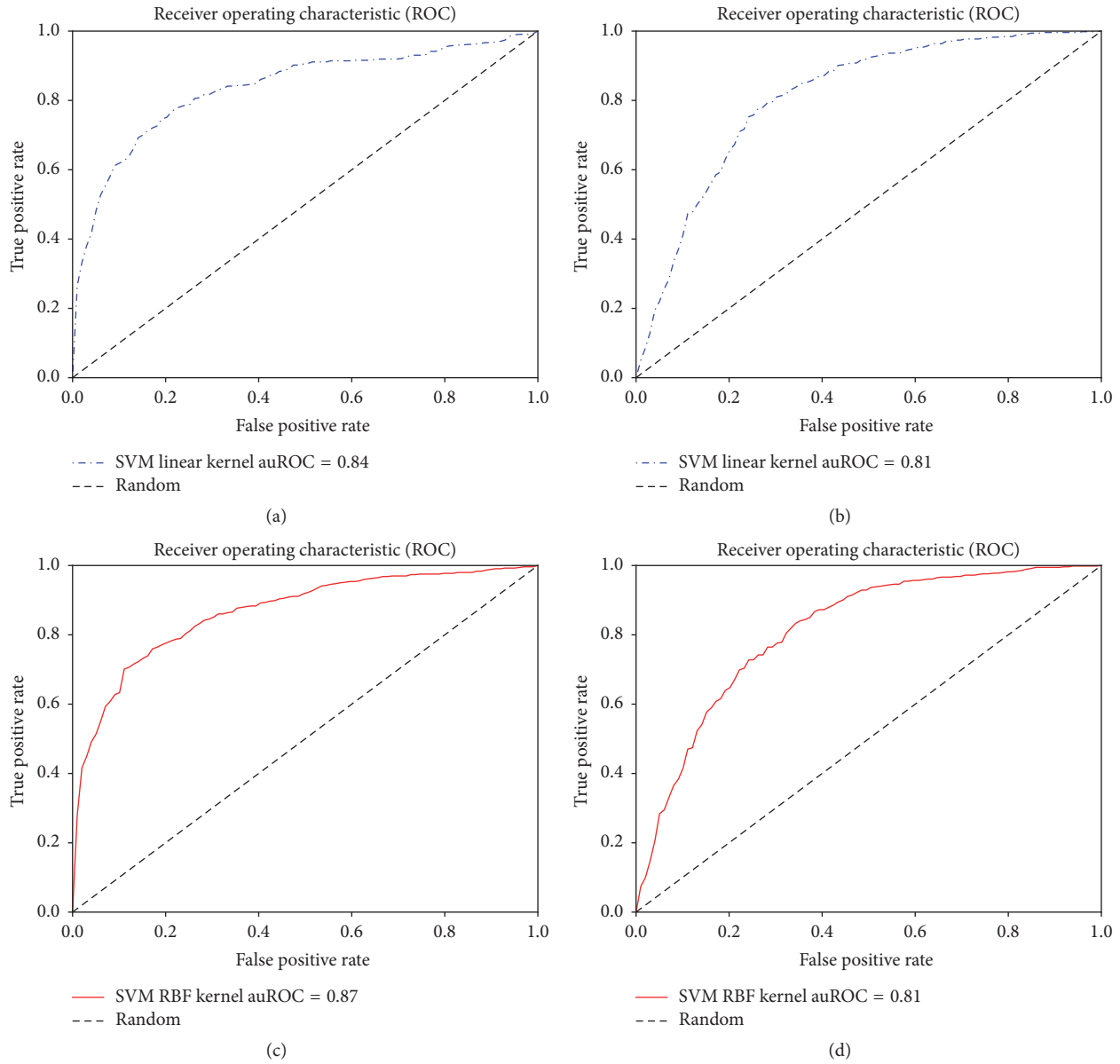

Figure 2: Using monogram features. Receiver operating characteristic curves for (a) SVM linear kernel classifier using HMM-Monogram features, (b) SVM linear kernel classifier using PSSM-Monogram features, (c) SVM RBF kernel classifier using HMM-Monogram features, and (d) SVM RBF kernel classifier using PSSM-Monogram features.

best results were found using the combination of HMM-Monogram and Bigram features and 82.87% accuracy was achieved using SVM linear kernels. In each case of the SVM linear kernel, HMM based features achieved better accuracy compared to PSSM based features. Similar results could be noticed for auROC, MCC, and sensitivity analysis. Specificity, auROC, and auPR are slightly improved in the experiments with SVM with RBF kernels. We also show the ROC curves for each of these experiments in Figures 2, 3, and 4.

3.2. Comparison with Other Methods. We have compared the performance of HMMBinder with several previous methods and tools used for DNA-binding protein prediction on the benchmark dataset *benchmark1075*. They are DNABinder [7], DNA-Prot [16], iDNA-Prot [11], iDNA-Prot|dis [14], DBP-Pred [17], iDNAPro-PseAAC [8], PseDNA-Pro [18], Kmer1 + ACC [19], and Local-DPP [20]. The results reported in this paper for these methods are taken from [8, 20]. The comparisons were made in terms of accuracy, sensitivity, specificity, MCC, and auROC. To make a fair comparison with the other methods, we performed jack-knife test as done in earlier studies and the results are reported in Table 2.

The best values in Table 2 are shown in bold faced fonts. The results show a clear margin of more than 7% improvement of accuracy over the previous best method, Local-DPP [20]. Similar improvements were found in other

Table 2: Comparison of performance of the proposed method with other state-of-the-art predictors using jack-knife test on the benchmark dataset.

Method	Accuracy	Sensitivity	Specificity	MCC	auROC
iDNAPro-PseAAC	76.76%	0.7562	0.7745	0.53	0.8392
DNABinder (dimension 21)	73.95%	0.6857	0.7909	0.48	0.8140
DNABinder (dimension 400)	73.58%	0.6647	0.8036	0.47	0.8150
DNA-Prot	72.55%	0.8267	0.5976	0.44	0.7890
iDNA-Prot	75.40%	0.8381	0.6473	0.50	0.7610
iDNA-Prot|dis	77.30%	0.7940	0.7527	0.54	0.8310
PseDNA-Pro	76.55%	0.7961	0.7363	0.53	—
Kmer1 + ACC	75.23%	0.7676	0.7376	0.50	0.8280
Local-DPP	79.20%	0.8400	0.7450	0.59	—
HMMBinder	**86.33%**	**0.8707**	**0.8555**	**0.72**	**0.9026**

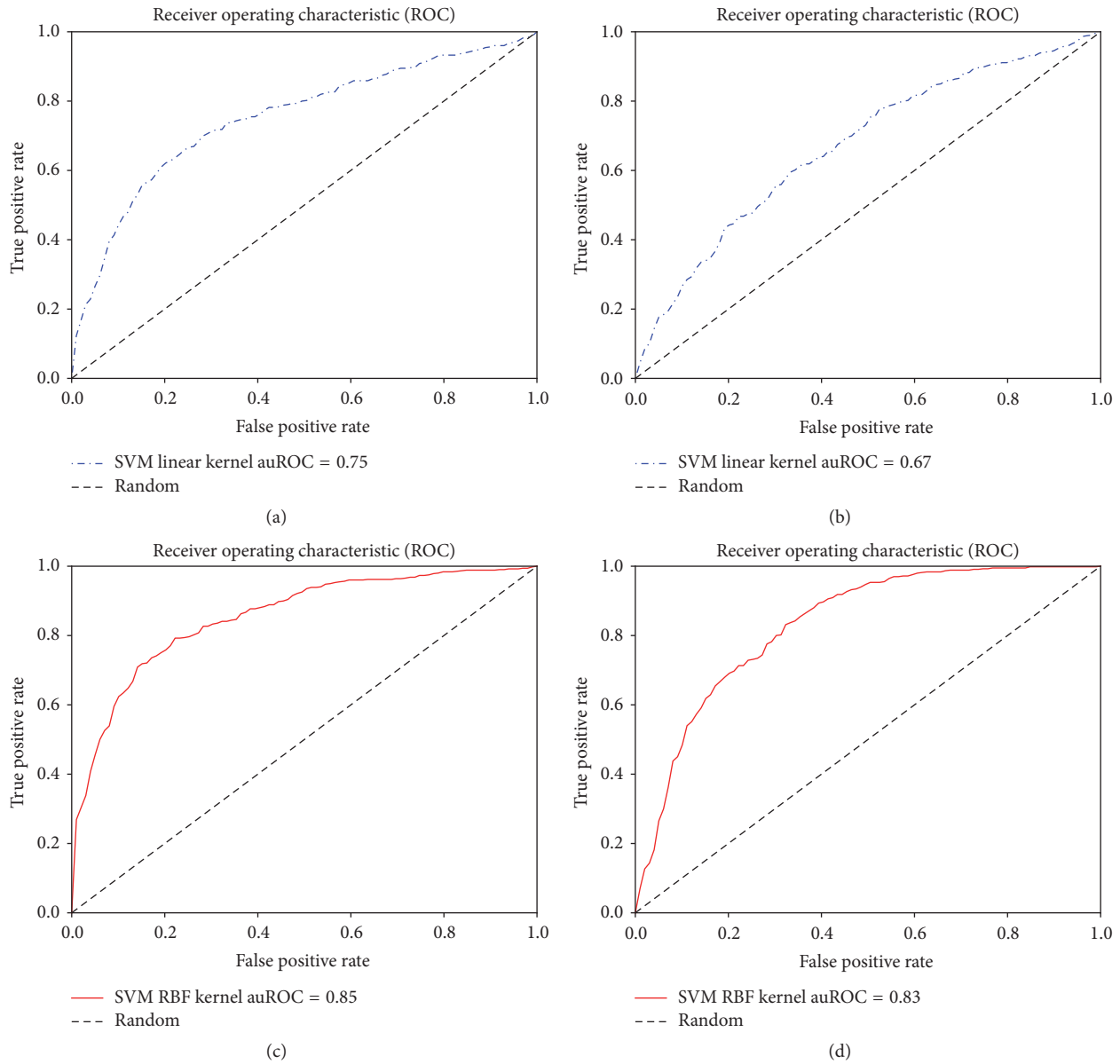

Figure 3: Using bigram features. Receiver operating characteristic curves for (a) SVM linear kernel classifier using HMM-Bigram features, (b) SVM linear kernel classifier using PSSM-Bigram features, (c) SVM RBF kernel classifier using HMM-Bigram features, and (d) SVM RBF kernel classifier using PSSM-Bigram features.

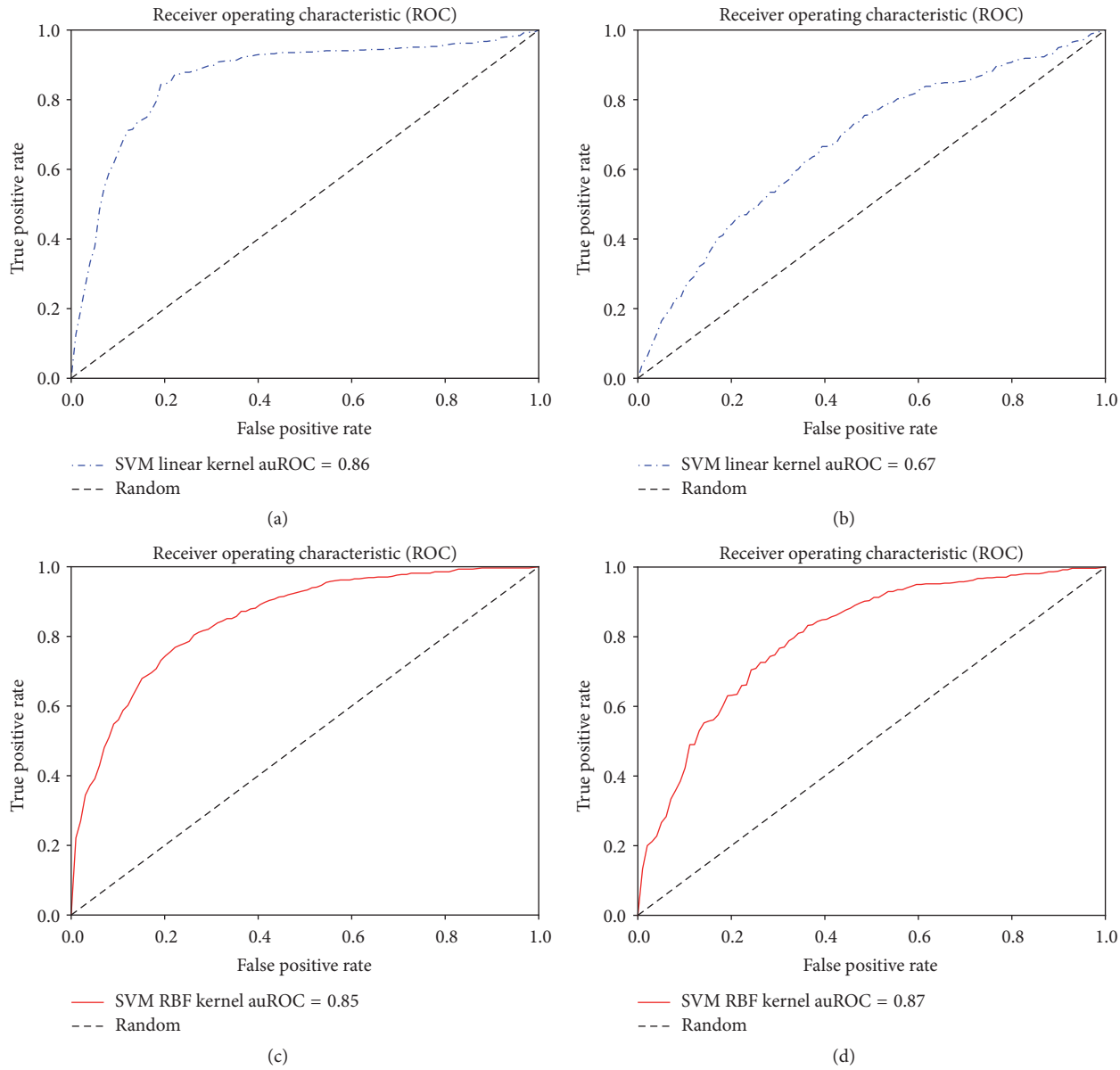

Figure 4: Using (Mono + Bi)gram features. Receiver operating characteristic curves for (a) SVM linear kernel classifier using HMM-Mono + Bigram features, (b) SVM linear kernel classifier using PSSM-Mono + Bigram features, (c) SVM RBF kernel classifier using HMM-Mono + Bigram features, and (d) SVM RBF kernel classifier using PSSM-Mono + Bigram features.

metrics too. Particularly, MCC is increased by 22% compared to the previous best method.

We further experimented to test the effectiveness of HMMBinder on the independent test set also. These results are shown in Table 3. Here the results are not the best but among the best. In terms of accuracy, our results are almost similar to iDNAPro-PseAAC [8]. Their results were significant in the benchmark dataset and were similar to ours in the independent dataset. Specificity value of HMMBinder was among the best and only second to DNA-Threader which failed miserably in terms of accuracy. Considering the difficulty level of the independent dataset, we believe that our method has not been overtrained on the benchmark dataset and the performance is promising and can be claimed as a generalized method after training and testing. Based on these results, we decided to build the web application based on the model trained on the benchmark dataset.

Note that the results on the independent dataset are comparative but not improved in comparison to the state-of-the-art methods. The main focus of this research was to build a classifier based on HMM profiles instead of the PSSM profile based features and we experimentally showed the effectiveness of the HMM profile based features over PSSM. In the future, we aim to focus on the independent dataset to perform better.

Additionally, we would like to highlight two points. Firstly, the datasets that we used were filtered using BLAST-CLUST. It is important to remove the sequences with

Table 3: Comparison of performance of the proposed method with other state-of-the-art predictors on the independent dataset.

Method	Accuracy	Sensitivity	Specificity	MCC	auROC
iDNAPro-PseAAC	69.89%	0.7741	0.6237	0.402	0.7754
iDNA-Prot	67.20%	0.6770	0.6670	0.344	—
DNA-Prot	61.80%	0.6990	0.5380	0.240	—
DNABinder	60.80%	0.5700	0.6450	0.216	0.6070
DNABIND	67.70%	0.6670	0.6880	0.355	0.6940
DNA-Threader	59.70%	0.2370	**0.9570**	0.279	—
DBPPred	76.90%	0.7960	0.7420	0.538	0.7910
iDNA-Prot|dis	72.00%	0.7950	0.6450	0.445	**0.7860**
Kmer1 + ACC	70.96%	0.8279	0.5913	0.431	0.7520
Local-DPP	**79.00**%	**0.9250**	0.6560	**0.625**	—
HMMBinder	69.02%	0.6153	0.7634	0.394	0.6324

similarity more than 25% from the dataset before applying the training and testing methods. We used the dataset proposed by Lou et al. [17], a widely accepted standard independent test dataset where the sequences with similarity of 25% or more with other sequences had been removed. We believe it would be interesting to see the effects of the other heuristic, CLUSTALW [47]. Secondly, feature selection methods are gaining much popularity in case of bioinformatics data and supervised machine learning. We believe that using sophisticated feature selection methods, such as maximum relevance minimum redundancy (mRMR) [48] and maximum relevance maximum distance (MRMD) [49], could improve the results further.

3.3. Web Server Implementation. We have implemented a web based application based on the proposed method. We call this HMMBinder. This is readily available to use at http://brl.uiu.ac.bd/HMMBinder. The server was implemented using PHP web programming language in the front end and python based prediction engine at the backend. The software requires an HMM profile as input to the tools that can be generated by HHBlits. The features are extracted automatically by the python program and the predicted value from a trained model is shown in the web form. The web site contains a "read me" guide and the necessary information required to run the application.

4. Conclusion

In this paper, we have introduced HMMBinder, a HMM profile based method for the DNA-binding protein prediction problem. We have used monogram and bigram features extracted from the HMM profiles generated by HHBlits and a SVM classification algorithm to train our data on a standard benchmark dataset. Our method is able to make considerable improvement over the other state-of-the-art methods on this dataset and performed comparably well in the independent dataset. We have also established a web based application for our method that is trained on the benchmark dataset. In the future, we wish to extract more effective features and generate larger dataset to train our model to be able to improve the results on the independent dataset. We believe there is a scope of improvement.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

References

[1] H. M. Berman, J. M. Thornton, N. M. Luscombe, and S. E. Austin, "An overview of the structures of protein-dna complexes," *Genome Biology*, vol. 1, 2000.

[2] E. W. Stawiski, L. M. Gregoret, and Y. Mandel-Gutfreund, "Annotating nucleic acid-binding function based on protein structure," *Journal of Molecular Biology*, vol. 326, no. 4, pp. 1065–1079, 2003.

[3] S. Jones, J. M. Thornton, H. P. Shanahan, and M. A. Garcia, "Identifying DNA-binding proteins using structural motifs and the electrostatic potential," *Nucleic Acids Research*, vol. 32, no. 16, pp. 4732–4741, 2004.

[4] R. Jaiswal, S. K. Singh, D. Bastia, and C. R. Escalante, "Crystallization and preliminary X-ray characterization of the eukaryotic replication terminator Reb1-Ter DNA complex," *Acta Crystallographica Section F:Structural Biology Communications*, vol. 71, pp. 414–418, 2015.

[5] R. E. Langlois and H. Lu, "Boosting the prediction and understanding of DNA-binding domains from sequence," *Nucleic Acids Research*, vol. 38, no. 10, Article ID gkq061, pp. 3149–3158, 2010.

[6] S. Ahmad, M. M. Gromiha, and A. Sarai, "Analysis and prediction of DNA-binding proteins and their binding residues based on composition, sequence and structural information," *Bioinformatics*, vol. 20, no. 4, pp. 477–486, 2004.

[7] M. Kumar, M. M. Gromiha, and G. P. S. Raghava, "Identification of DNA-binding proteins using support vector machines and evolutionary profiles," *BMC Bioinformatics*, vol. 8, no. 1, article 463, 2007.

[8] B. Liu, S. Wang, and X. Wang, "DNA binding protein identification by combining pseudo amino acid composition and profile-based protein representation," *Scientific Reports*, vol. 5, Article ID 15479, 2015.

[9] L. Song, D. Li, X. Zeng, Y. Wu, L. Guo, and Q. Zou, "nDNA-prot: identification of DNA-binding proteins based on unbalanced classification," *BMC Bioinformatics*, vol. 15, no. 1, article 298, 2014.

[10] C. Yan, M. Terribilini, F. Wu, R. L. Jernigan, D. Dobbs, and V. Honavar, "Predicting DNA-binding sites of proteins from amino acid sequence," *BMC Bioinformatics*, vol. 7, no. 1, article 262, 2006.

[11] W.-Z. Lin, J.-A. Fang, X. Xiao, and K.-C. Chou, "iDNA-prot: identification of DNA binding proteins using random forest with grey model," *PLoS ONE*, vol. 6, no. 9, Article ID e24756, 2011.

[12] J. Zhou, Q. Lu, R. Xu, L. Gui, and H. Wang, "CNNsite: Prediction of DNA-binding residues in proteins using Convolutional Neural Network with sequence features," in *Proceedings of the 2016 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2016*, pp. 78–85, China, December 2016.

[13] A. Szilágyi and J. Skolnick, "Efficient Prediction of Nucleic Acid Binding Function from Low-resolution Protein Structures," *Journal of Molecular Biology*, vol. 358, no. 3, pp. 922–933, 2006.

[14] B. Liu, J. Xu, and X. Lan, "iDNA-Prot—dis: identifying dna-binding proteins by incorporating amino acid distance-pairs and reduced alphabet profile into the general pseudo amino acid composition," *PLoS ONE*, vol. 9, no. 9, Article ID e106691, 2014.

[15] Y. Fang, Y. Guo, Y. Feng, and M. Li, "Predicting DNA-binding proteins: approached from Chou's pseudo amino acid composition and other specific sequence features," *Amino Acids*, vol. 34, no. 1, pp. 103–109, 2008.

[16] K. K. Kumar, G. Pugalenthi, and P. N. Suganthan, "DNA-prot: identification of DNA binding proteins from protein sequence information using random forest," *Journal of Biomolecular Structure and Dynamics*, vol. 26, no. 6, pp. 679–686, 2009.

[17] W. Lou, X. Wang, F. Chen, Y. Chen, B. Jiang, and H. Zhang, "Sequence based prediction of DNA-binding proteins based on hybrid feature selection using random forest and Gaussian naïve Bayes," *PLoS ONE*, vol. 9, no. 1, Article ID e86703, 2014.

[18] B. Liu, J. Xu, S. Fan, R. Xu, J. Zhou, and X. Wang, "PseDNA-Pro: DNA-binding protein identification by combining Chou's PseAAC and physicochemical distance transformation," *Molecular Informatics*, vol. 34, no. 1, pp. 8–17, 2015.

[19] Q. Dong, S. Wang, K. Wang, X. Liu, and B. Liu, "Identification of DNA-binding proteins by auto-cross covariance transformation," in *Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2015*, pp. 470–475, USA, November 2015.

[20] L. Wei, J. Tang, and Q. Zou, "Local-DPP: an improved DNA-binding protein prediction method by exploring local evolutionary information," *Information Sciences*, vol. 384, pp. 135–144, 2017.

[21] R. Xu, J. Zhou, H. Wang, Y. He, X. Wang, and B. Liu, "Identifying DNA-binding proteins by combining support vector machine and PSSM distance transformation," *BMC Systems Biology*, vol. 9, no. 1, article S10, 2015.

[22] J. Im, N. Tuvshinjargal, B. Park, W. Lee, D.-S. Huang, and K. Han, "PNImodeler: web server for inferring protein-binding nucleotides from sequence data," *BMC Genomics*, vol. 16, no. 3, article S6, 2015.

[23] I. Paz, E. Kligun, B. Bengad, and Y. Mandel-Gutfreund, "BindUP: a web server for non-homology-based prediction of DNA and RNA binding proteins," *Nucleic Acids Research*, vol. 44, no. W1, pp. W568–W574, 2016.

[24] H. P. Shanahan, M. A. Garcia, S. Jones, and J. M. Thornton, "Identifying DNA-binding proteins using structural motifs and the electrostatic potential," *Nucleic Acids Research*, vol. 32, no. 16, pp. 4732–4741, 2004.

[25] G. Nimrod, M. Schushan, A. Szilágyi, C. Leslie, and N. Ben-Tal, "iDBPs: a web server for the identification of DNA binding proteins," *Bioinformatics*, vol. 26, no. 5, Article ID btq019, pp. 692-693, 2010.

[26] R. Xu, J. Zhou, B. Liu et al., "Identification of DNA-binding proteins by incorporating evolutionary information into pseudo amino acid composition via the top-n-gram approach," *Journal of Biomolecular Structure and Dynamics*, vol. 33, no. 8, pp. 1720–1730, 2015.

[27] X.-W. Zhao, X.-T. Li, Z.-Q. Ma, and M.-H. Yin, "Identify DNA-binding proteins with optimal Chou's amino acid composition," *Protein and Peptide Letters*, vol. 19, no. 4, pp. 398–405, 2012.

[28] J. Lyons, A. Dehzangi, R. Heffernan et al., "Advancing the accuracy of protein fold recognition by utilizing profiles from hidden Markov models," *IEEE Transactions on NanoBioscience*, vol. 14, no. 7, pp. 761–772, 2015.

[29] K.-C. Chou, "Some remarks on protein attribute prediction and pseudo amino acid composition," *Journal of Theoretical Biology*, vol. 273, pp. 236–247, 2011.

[30] M. Remmert, A. Biegert, A. Hauser, and J. Söding, "HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment," *Nature Methods*, vol. 9, no. 2, pp. 173–175, 2012.

[31] B. Liu, L. Fang, F. Liu et al., "Identification of real microRNA precursors with a pseudo structure status composition approach," *PLoS ONE*, vol. 10, no. 3, Article ID e0121501, 2015.

[32] H. M. Berman, J. Westbrook, Z. Feng et al., "The protein data bank," in *International Tables for Crystallography Volume F: Crystallography of biological macromolecules*, pp. 675–684, 2006.

[33] I. Dondoshansky and Y. Wolf, *Blastclust (NCBI Software Development Toolkit)*, NCBI, Maryland, Md, USA, 2002.

[34] D. B. Kuchibhatla, W. A. Sherman, B. Y. W. Chung et al., "Powerful sequence similarity search methods and in-depth manual analyses can identify remote homologs in many apparently 'orphan' viral proteins," *Journal of Virology*, vol. 88, no. 1, pp. 10–20, 2014.

[35] UniProt Consortium et al., "Uniprot: the universal protein knowledgebase," *Nucleic Acids Research*, vol. 45, no. D1, pp. D158–D169, 2017.

[36] Y.-H. Taguchi and M. M. Gromiha, "Application of amino acid occurrence for discriminating different folding types of globular proteins," *BMC Bioinformatics*, vol. 8, no. 1, article 404, 2007.

[37] A. Sharma, J. Lyons, A. Dehzangi, and K. K. Paliwal, "A feature extraction technique using bi-gram probabilities of position specific scoring matrix for protein fold recognition," *Journal of Theoretical Biology*, vol. 320, pp. 41–46, 2013.

[38] S. F. Altschul, T. L. Madden, A. A. Schäffer et al., "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs," *Nucleic Acids Research*, vol. 25, no. 17, pp. 3389–3402, 1997.

[39] R. Sharma, A. Dehzangi, J. Lyons, K. Paliwal, T. Tsunoda, and A. Sharma, "Predict gram-positive and gram-negative subcellular localization via incorporating evolutionary information and physicochemical features into chou's general PseAAC," *IEEE Transactions on NanoBioscience*, vol. 14, no. 8, pp. 915–926, 2015.

[40] A. Sharma, K. K. Paliwal, A. Dehzangi, J. Lyons, S. Imoto, and S. Miyano, "A strategy to select suitable physicochemical attributes of amino acids for protein fold recognition," *BMC Bioinformatics*, vol. 14, no. 1, article 233, 2013.

[41] A. Dehzangi, S. Sohrabi, R. Heffernan et al., "Gram-positive and gram-negative subcellular localization using rotation forest and physicochemical-based features," *BMC Bioinformatics*, vol. 16, no. 4, article S1, 2015.

[42] A. Dehzangi, A. Sharma, J. Lyons, K. K. Paliwal, and A. Sattar, "A mixture of physicochemical and evolutionarybased feature extraction approaches for protein fold recognition," *International Journal of Data Mining and Bioinformatics*, vol. 11, no. 1, pp. 115–138, 2014.

[43] A. Dehzangi, K. Paliwal, J. Lyons, A. Sharma, and A. Sattar, "Enhancing protein fold prediction accuracy using evolutionary and structural features," in *Proceedings of the IAPR International Conference on Pattern Recognition in Bioinformatics*, pp. 196–207, Springer, Berlin, Germany, 2013.

[44] D. M. Powers, "Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation," 2011.

[45] B. Efron and G. Gong, "A leisurely look at the bootstrap, the jackknife, and cross-validation," *The American Statistician*, vol. 37, no. 1, pp. 36–48, 1983.

[46] F. Pedregosa, G. Varoquaux, A. Gramfort et al., "Scikit-learn: machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[47] J. D. Thompson, T. J. Gibson, D. G. Higgins et al., "Multiple sequence alignment using ClustalW and ClustalX," *Current Protocols in Bioinformatics*, pp. 2-3, 2002.

[48] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.

[49] Q. Zou, J. Zeng, L. Cao, and R. Ji, "A novel features ranking metric with application to scalable visual and bioinformatics data classification," *Neurocomputing*, vol. 173, part 2, pp. 346–354, 2016.